Computational Complexity Theory, Fall 2008 September 26

Lecture 8: MULTIPLICATION $\in$ TC^0

Lecturer: Kristoffer Arnsfelt Hansen Scribe: Michael Kølbæk Madsen

1 Lecture 8

In this lecture it's shown that MULTIPLICATION is in TC^0. The first part will be the definition of threshold circuits and the second part will be a constructive proof for multiplication is in TC^0

1.1 Majority and TC^0

Definition 1 (MAJORITY) MAJORITY is shortly named MAJ.

$$\mathrm{MAJ}((x_1,\ldots,x_n)) = \begin{cases} 1 & \text{if} \sum_{x_i}^{n} \geq \frac{n}{2} \\ 0 & \text{otherwise} \end{cases}$$

Definition 2 (TC^0) *The class of polynomial size, constant depth, unbounded fanin circuits built using MAJORITY gates.*

Proposition 3 ($AC^0 \subseteq TC^0$) **Proof** AC^0 *is the class of languages decided by constant depth, polynomial size circuits, with unbounded fanin. Since* AC^0 *only consist of AND and OR gates, it's simple to simulate it using MAJORITY gates and constants.*

$$AND(x_1,\ldots,x_n) = MAJ(x_1,\ldots,x_n,\overbrace{0,\ldots,0}^{n}) \tag{1}$$

$$OR(x_1,\ldots,x_n) = MAJ(x_1,\ldots,x_n,\overbrace{1,\ldots,1}^{n-1}). \tag{2}$$

The usage of majority gates doesn't change the depth nor the size of the circuit. □

Definition 4 (Oracle circuits) *An oracle circuit is a circuit with oracle gates* A. $AC^0[A]$ *is* AC^0 *circuits with additional* A *gates. A gate from* A *has depth* 1 *and size* t *where* t *is the number of inputs to the gate.*

TC^0 can be defined as $AC^0[MAJ]$.

Definition 5 (Threshold)

$$T_k(x_1,\ldots,x_n) = \begin{cases} 1 & \textit{if } \sum_{i=1}^{n} \geq k \\ 0 & \textit{otherwise} \end{cases} \tag{3}$$

It's fortunately easy to do with MAJORITY.

$$T_k(x_1,\ldots,x_n) = \begin{cases} MAJ(x_1,\ldots,x_n,\overbrace{1,\ldots,1}^{\frac{2k-n}{4}}) & \textit{if } k \geq \frac{n}{2} \\ MAJ(x_1,\ldots,x_n,\overbrace{0,\ldots,0}^{\frac{n-2k}{4}}) & \textit{otherwise} \end{cases} \tag{4}$$

1.2 Multiplication is in TC^0

Definition 6 (ITADD) *Given n n-bit numbers $a_1, \ldots, a_n$ $ITADD(a_1, \ldots, a_2) = \sum_{i=0}^{n} a_i$.*

Theorem 7 *$ITADD \in \mathrm{NC}^1$*

Proof The first thing to prove is that it's possible to add 3 n bit numbers and get one n+2 bit number Let a, b, c be n-bit numbers and d,e $n+1$ bit numbers. e is going to hold the carry, and d the bitwise addition of a,b and c.

$$\begin{aligned} a &= (a_{n-1}, \ldots, a_0) \\ b &= (b_{n-1}, \ldots, b_0) \\ c &= (c_{n-1}, \ldots, c_0) \\ d &= (d_n, \ldots, d_0) \\ e &= (e_n, \ldots, e_0) \\ e_0 &= 0 \\ e_i &= T_2(a_{i-1}, b_{i-1}, c_{i-1}) \\ d_i &= a_i \oplus b_i \oplus c_i \\ d_n &= 0 \end{aligned}$$

The tripple $a+b+c$ is reduced to an addition of 2 $n+1$-bit numbers. It's possible to do this for all tripples. Which then gives $\frac{2n}{3}$ $n+1$-bit numbers. After $O(\log n)$ iterations there's 2 $n + O(\log n)$-bits numbers left which can be added using a constant depth AC^0 circuit. But we had to make $O(\log n)$ iterations of these three to two number reduction so the resulting circuit will be the height of $O(\log n)$ stacked NC^0 circuits. with the last addition as a AC^0 circuit. Which clearly is in NC^1. □

Proposition 8 ($\mathrm{TC}^0 \subseteq \mathrm{NC}^1$) **Proof** TC^0 *consist of unbounded fanin threshold gates, but each of theese gates can be swapped out with a $O(\log n)$ deep NC^1 circuit. Since a TC^0 circuit has constant depth the resulting NC^1 circuit would have $O(\log n)$ depth and bounded fanin. The MAJORITY gates can be constructed using the iterated addition method mentioned above.* □

Definition 9 (MULTIPLICATION) *Given 2 n-bit numbers a, b calculate $a \cdot b = c$ where c is a $2n$-bit number.*

Definition 10 (Constant depth reduction) *When we are working in logspace it makes no sence to use logspace reductions, because then it would be possible to solve the problem in constant space. Instead we are considering constant depth reductions $\leq_{cd}$.* $\mathrm{A} \leq_{cd} \mathrm{DifA} \in \mathrm{AC}^0[\mathrm{D}]$. *Ie. if* $\mathrm{B} \in \mathrm{NC}^1$ *and* $\mathrm{A} \leq_{cd} \mathrm{B}$ *then* $\mathrm{A} \in \mathrm{NC}^1$ *Remark: Since* $\mathrm{TC}^0 = \mathrm{AC}^0[\mathrm{MAJ}]$. *MAJ* $\in \mathrm{NC}^1 \Rightarrow \mathrm{TC}^0 \in \mathrm{NC}^1$.

Theorem 11 (MULTIPLICATION $\leq_{\mathbf{cd}}$ ITADD) **Proof** *Using longhand multiplication it's just an addition of n n-bit numbers.* □

Definition 12 (BCOUNT) BCOUNT *is bitcount. Given n bits $a_1, \ldots, a_n$ calculate $s = \sum_{i=1}^{n} a_i$. where $s = (s_k, \ldots, s_0)$, $k = \lfloor \log n \rfloor$.* $\mathrm{BCOUNT} \in \mathrm{TC}^0$.

Definition 13 (LOG − ITADD) *Given* $\log n$ *n-bit numbers* $a^1, \ldots, a^{\log n}$*, calculate the sum* $\sum_{i=0}^{\log n} a_i$*.*

Proposition 14 (ITADD $\in$ TC^0) *Given* n *n-bit* $a^1, \ldots, a^n$*. Calculate* $\sum_{i=1}^{n} a^i$*. The proof uses* BCOUNT. **Proof**

$$R_i = \{j \in \{1, \ldots, n\} | \textit{The i'th bit of } j \textit{ is } 1\} \quad (5)$$

$$s_i = \bigvee_{j \in R_i} (T_j(a_1, \ldots, a_n) \wedge \neg T_{j+1}(a_1, \ldots, a_n)) \quad (6)$$

In words $T_j(a_1, \ldots, a_n)$ *is* 1 *if the sum is at least* j*.*

Let $lg(n) = \lfloor \log_2 n + 1 \rfloor$

$$\begin{array}{c} (a^1_{n-1}, \quad \ldots, \quad a^1_0) \\ \vdots \\ (a^n_{n-1}, \quad \ldots, \quad a^n_0) \\ \hline s^{n-1}, \quad \ldots, \quad s^0 \end{array}$$

Where $s^i = (s^i_{\log n - 1}, \ldots, s^i_0)$ *is the bitcount for the i'th column.*

$$\sum_{1=1}^{n} a^i = \sum_{i=0}^{n} s^i \dot{2}^i \quad (7)$$

$$= \sum_{i=0}^{n} \sum_{j=0}^{\log n - 1} s^i_j \cdot 2^j \cdot 2^i \quad (8)$$

$$= \sum_{j=0}^{\log n - 1} (\sum_{i=0}^{n} s^i_j \cdot 2^{i+j}) \quad (9)$$

Which means, add n*,* n*-bit numbers is now reduced to add* $lg\ n$*,* $n + lg\ n$*-bit numbers.*
In the same way it's possible to continue.
1. lgn*,* n*-bit numbers can be done adding* $lg^{(2)}n$*,* $(n + lgn)$*-bit numbers.*
2. This reduction is can be done until $lg^{(k)}n \leq 2$*. Then there's* $lg^{(k)}n$*,* $(n + lg^{(1)}n + \cdots + lg^{(k-1)}n)$*-bit numbers.*
One output bit of the output depends only on $O(\log n)$ *bits of the input. The height of the total circuit adding* $\log n$ *n-bit numbers, is* $lg^{(2)}n \cdot lg^{(3)}n \cdots lg^{(k)}n = O(\log n)$*. Which means it can be done in* AC^0*.*
3. Then the only missing part is to add the last 2 numbers with an AC^0 *circuit.*
This means ITADD *is in* TC^0 *(since we did use* TC^0 *circuits for the bitcount). But since* ITADD $\in$ TC^0 *and* MULTIPLICATION $\leq_{cd}$ ITADD *then* MULTIPLICATION $\in$ TC^0*.* □

1.3 Limits for AC^0

Definition 15 (PARITY)

$$\text{PARITY}(\text{x}_1, \ldots, \text{x}_\text{n}) = \begin{cases} 1 & \textit{if } (\sum_{i=1}^{n} x_i) \in \{2z + 1 | z \in \mathbb{N}\} \\ 0 & \textit{otherwise} \end{cases} \quad (10)$$

$$= x_1 \oplus x_2 \oplus \cdots \oplus x_n \quad (11)$$

If it's possible to show that PARITY $\notin \mathrm{AC}^0$ then MULTIPLICATION $\notin \mathrm{AC}^0$ since PARITY $\leq_{\mathrm{cd}}$ MULTIPLICATION. To do this we need the switching lemma, which will not be proved in this note.